USING RULES AND TASK DIVISION TO AUGMENT CONNECTIONIST LEARNING

Technical Report AIP - 36

W. Oliver & W. Schneider

Learning Research and Development Center
University of Pittsburgh
Pittsburgh, PA. 15260

The Artificial Intelligence and Psychology Project

Departments of
Computer Science and Psychology
Carnegie Mellon University

Learning Research and Development Center
University of Pittsburgh

2

USING RULES AND TASK DIVISION TO AUGMENT CONNECTIONIST LEARNING

Technical Report AIP - 36

W. Oliver & W. Schneider

Learning Research and Development Center
University of Pittsburgh
Pittsburgh, PA. 15260

July 1988

Codes
/or

A-1

This research was supported by the Computer Sciences Division, Office of Naval Research and DARPA under Contract Number N00014-86-K-0678; Reproduction in whole or in part is permitted for purposes of the United States Government. Approved for public release; distribution unlimited.

Unclassi. ied

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION Unclassified	1b RESTRICTIVE MARKINGS
2a. SECURITY CLASSIFICATION AUTHORITY	3 DISTRIBUTION / AVAILABILITY OF REPORT Approved for public release; Distribution unlimited
2b. DECLASSIFICATION / DOWNGRADING SCHEDULE	
4. PERFORMING ORGANIZATION REPORT NUMBER(S) AIP - 36	5. MONITORING ORGANIZATION REPORT NUMBER(S)
6a. NAME OF PERFORMING ORGANIZATION Carnegie-Mellon University 6b OFFICE SYMBOL (If applicable)	7a NAME OF MONITORING ORGANIZATION Computer Sciences Division Office of Naval Research
6c. ADDRESS (City, State, and ZIP Code) Department of Psychology Pittsburgh, Pennsylvania 15213	7b ADDRESS (City, State, and ZIP Code) 800 N. Quincy Street Arlington, Virginia 22217-5000
8a. NAME OF FUNDING / SPONSORING ORGANIZATION Same as Monitoring Organization 8b OFFICE SYMBOL (If applicable)	9 PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER N00014-86-K-0678
8c. ADDRESS (City, State, and ZIP Code)	10 SOURCE OF FUNDING NUMBERS p4000ub201/7-4-86

PROGRAM ELEMENT NO	PROJECT NO.	TASK NO.	WORK UNIT ACCESSION NO
N/A	N/A	N/A	N/A

11 TITLE (Include Security Classification)
Using rules and task division to augment connectionist learning

12 PERSONAL AUTHOR(S)
Oliver, William L. and Schneider, Walter

13a. TYPE OF REPORT	13b. TIME COVERED	14 DATE OF REPORT (Year, Month, Day)	15. PAGE COUNT
Technical	FROM 86Sept15 TO 91Sept14	1988 July 1	11

16 SUPPLEMENTARY NOTATION
submitted for publication to Proceedings of the Tenth Annual Conference of the Cognitive Science Society

17 COSATI CODES			18 SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	neural models, connectionism scaling, learning. EG

19 ABSTRACT (Continue on reverse if necessary and identify by block number)
See reverse side

Cont From
1473

20 DISTRIBUTION / AVAILABILITY OF ABSTRACT ☐ UNCLASSIFIED/UNLIMITED ☒ SAME AS RPT ☐ DTIC USERS	21 ABSTRACT SECURITY CLASSIFICATION
22a NAME OF RESPONSIBLE INDIVIDUAL Dr. Alan L. Meyrowitz	22b TELEPHONE (Include Area Code) (202) 696-4302 22c OFFICE SYMBOL N00014

DD FORM 1473, 84 MAR 83 APR edition may be used until exhausted. All other editions are obsolete

SECURITY CLASSIFICATION OF THIS PAGE

Using Rules and Task Division to Augment Connectionist Learning

William L. Oliver and Walter Schneider

Learning Research and Development Center
University of Pittsburgh

Abstract

Learning as a function of task complexity was examined in human learning and two connectionist simulations. An example task involved learning to map basic input/output digital logic functions for six digital gates (AND OR, XOR and negated versions) with 2- or 6-inputs. Humans given instruction learned the task in about 300 trials and showed no effect of the number of inputs. Backpropagation learning in a network with 20 hidden units required 68,000 trials and scaled poorly, requiring 8 times as many trials to learn the 6-input gates as to learn the 2-input gates. A second simulation combined backpropagation with task division based upon rules humans use to perform the task. The combined approach improved the scaling of the problem, learning in 3,100 trials and requiring about 3 times as many trials to learn the 6-input gates as to learn the 2-input gates. Issues regarding scaling and augmenting connectionist learning with rule-based instruction are discussed.

(cont 1473 on Front)

Introduction

In this paper we compare human learning of a modestly complex task with connectionist learning that used the procedure known as "backpropagation" (Rumelhart, Hinton & Williams, 1986). We also consider a model that uses rules to divide the task into subtasks that can be separately learned with backpropagation. We examine the benefits of providing a connectionist system with a rule-based instructor that can reconfigure the system via attention to learn components of the task.

A critical issue for artificial intelligence and human learning involves finding learning algorithms that scale well. Learning time for an algorithm should not increase so dramatically with task complexity that it can only be applied to toy problems. Minsky and Papert (1988, p. 262) comment on the importance of the scale issue stating: "In the examination of theories of learning and problem solving, the study of such growths in cost is not merely one more aspect to be taken into account; it is the only aspect worth considering."

To the psychologist the problem of scale has critical importance because the time a biological system has to learn is limited. A learning algorithm that does not allow the organism to learn a task in its lifetime is of limited value.

Current connectionist algorithms may scale too poorly to account for human learning in many instances. Many tasks may be learned far more quickly by humans than by currently available connectionist procedures, because human learning can be guided by rules. Below we describe such a task in which humans required around 300 trials to learn. In contrast, currently our fastest learning simulations using only backpropagation required 68,000 trials. (see Figure 2 below). More importantly, human learning time did not increase with increases in the complexity of the task, whereas the learning times for the connectionist procedure significantly increased.

The study of connectionist learning is partially supported by an implicit assumption that humans provide an existence proof for simple, powerful learning algorithms that scale well. This assumption is likely to be false. By *simple* learning algorithms we mean algorithms that can map inputs to outputs by altering connection weights on each trial given the input and the desired output state of the system. This learning occurs without using explicit rules or focus-

1473

ing the network's attention on specific parts of the problem. Human learning in such situations is poor and does not scale well. Subjects take many trials to learn simple concepts involving very few feature dimensions (usually about 4) in psychological studies in which subjects are discouraged from forming verbal rules (e.g., Medin and Schaffer, 1978). Humans benefit greatly from focusing attention, instruction, hypothesis generation, and learning by imitation, none of which is present in traditional connectionist learning models. When learning a complex problem, such as family hierarchies (Hinton, 1985), a connectionist procedure must develop internal representations solely from the inputs and outputs that are specified on each learning trial. There is no mechanism to directly instruct the network about relationships among features (e.g., that *female* and *daughter* are correlated features such that daughters are always female). The backpropagation procedure can learn simple tasks of this sort, but learning often requires thousands of trials. We believe that both simple learning algorithms and rule-based learning will be necessary to account for human learning.

The human learning of chicken sexing (identifying young chicks as males or females) provides a contrast between learning by input-output mapping and learning by instruction on rules. Until recently chicken sexers had to learn their task on the basis of feedback from experts and on-the-job practice. It was claimed to have taken years for people to become proficient at this task (Biederman & Shiffrar, 1987). Biederman and Shiffrar demonstrated that college students could perform a variant of the chicken sexing task as well as experts when provided with a classification rule. Only about a minute was needed to instruct the subjects on this rule, which focused subjects' attention on particular features and told them how to respond given the presence of those features. This example suggests that humans can learn complex relations via reinforced input-output mapping, but this learning method scales poorly and can be greatly improved by using attentional and instructional operations that are generally absent in connectionist learning.

We are examining connectionist architectures that include attentional focusing and instruction-based learning (Schneider & Detweiler,1987; Schneider & Mumme,1988; Schneider & Oliver, 1988). These architectures combine features from connectionist and production-system models. Rule-based processing allows an attentional mechanism to dynamically reconfigure connectionist networks so that critical features become salient and a task can be decomposed into subtasks of smaller scale. Using rules allows rapid initial learning of the components of the task and the serial execution of each component, as occurs in Anderson's (1983) ACT* or Laird, Rosenbloom and Newell's (1986) SOAR. Connectionist learning within the architecture can convert serial processing of the component rules to parallel processing as a consequence of practice. In addition, the mutual constraint nature of connectionist processing provides a best-match mapping of inputs to outputs that is less brittle than rule-based matching processes.

In this paper we examine the benefits of task decomposition by comparing the human learner to a connectionist learning system with and without task decomposition. We examine the effect of learning as a function of the complexity of the task. The task involved learning digital input-output mappings for six digital logic gates (AND, OR, XOR and the negated forms of the rules) for either 2, 4 or 6 inputs per gate. We have studied this task extensively in the acquisition of human troubleshooting skill (Carlson, Sullivan & Schneider, 1988a, 1988b). When learning this task, human subjects describe their processing as having three stages. The first stage is encoding the inputs as all 1's, all 0's, or mixed. The second stage is mapping the coded input and the gate type to the expected output of a 0 or 1. The third stage involves applying the negation operator when it is required to reverse the output. Subjects were instructed on rules for each stage and then required to learn 2-, 4-or 6- input gate problems. Connectionist learning without decomposition was examined in a network that mapped the inputs to the outputs through a single layer of hidden units. Input-output pairs were presented to the network, and backpropagation learning (Rumelhart et al., 1986) was used to modify the connection weights. Connectionist learning with decomposition was examined in a network

composed of three modules, one for each stage. Each module had an input layer and an output layer. During training, each module received input and output information for each stage and propagated error only within its own stage.

Human Learning of Digital Logic

The computational properties of connectionist models have been studied by examining how they learn boolean functions (e.g., Minsky & Papert, 1988; Rumelhart et al., 1986; Volper & Hampson, 1986). Interestingly, research on digital trouble shooting has also looked at how subjects learn boolean logic in the laboratory (Brooke & Duncan, 1983; Carlson et al., 1988a, 1988b). In order to compare a connectionist model's learning with human learning, we designed an experiment that required subjects to learn several boolean functions and later had the model learn the same set of functions. We were mainly interested in whether increasing the complexity of the task by increasing the number of inputs to the functions would make the task much more difficult to learn.

The subjects in this experiment were University of Pittsburgh undergraduates with no experience in digital logic. A between-subjects experimental design was used; one group of 8 subjects learned digital logic gates with 2 inputs and another group of 9 subjects learned gates with 6 inputs. The subjects' task was to learn the rules for the gates to a high level of accuracy while responding as quickly as possible. Subjects typically reach an asymptotic accuracy of only about 92% in this task (Carlson, et al., 1988b). Their errors are random, suggesting causes other than rule learning (e.g., attention shifts, speed-accuracy trade-offs) for the less-than-perfect performance.

The subjects learned six digital logic rules--AND, NAND, OR, NOR, XOR, XNOR. The subjects predicted the correct outputs when given different combinations of 0's and 1's as inputs for the various logic gates. The inputs to the gates were randomly determined with certain constraints on each trial (see below). The gates and their inputs appeared one at a time on a CRT screen, and the subjects indicated the correct output (0 or 1) by pressing labelled keys. A computer controlled the sequencing and presentation of the stimuli and gathered data on the accuracy and speed of the subjects' responses. Feedback on the correctness of response was provided after each trial. The subjects were given verbal rules during the early part of the experiment for each gate, such as the following rule for the AND gate: "if the the inputs are all 1's respond 1; if the inputs are mixed (0's and 1's) respond 0; and if the inputs are all 0's respond 0." When a help key was pressed, the appropriate rules for a gate appeared in the upper-left-hand corner of the screen. An introduction to the three gate types (AND, OR, and XOR) involving 24 trials per gate was followed by 36 practice trials responding to gates and inputs selected at random. The subjects were then given instructions on how to carry out negation for the different gates (NAND, NOR, and XNOR) and given 24 trials of practice on each of these gate types. An additional 36 practice trials followed in which the negated gates were selected at random and presented to the subjects. In the final part of the experiment, the subjects responded to 300 gates selected at random from the entire set, including negated gates. The subjects could rest briefly after blocks of 50 trials, and use of the help key was not permitted.

In order to vary the complexity of the task, the number of inputs to the gates differed between groups of subjects. One group of subjects saw gates with 2 inputs and another group saw gates with 6 inputs. Because increasing the number of inputs dramatically changes the proportion of 1 and 0 responses for a given gate, a constraint was placed on the sampling of input combinations for the 6-input condition. For the 6-input gates, the probability of sampling certain input combinations (e.g., the all 1's case for the AND gate) was increased to maintain the same proportions of 0 and 1 responses as occurred in the 2-input condition. Without this constraint on the generation of input combinations, the subjects would be biased towards always giving the same response for a particular gate--for example, they would be biased

towards responding 0 to every AND gate because the probability of that answer being correct would be .98.

Human Learning Results

The subjects responded correctly on a high proportion of trials (92%) during the final 300 trials of practice. The mean percentages of correct responses over 50-trial blocks were 89, 90, 94, 95, 94, and 93% for blocks 1 through 6 respectively. Hence, the subjects started this final part of the experiment with high accuracy and became somewhat more accurate with the additional practice. An analysis of variance that included the variables for input condition and 50-trial blocks indicated that there were significant differences in accuracy among the blocks, $F(5,75)=4.60$, $p<.001$. The main effect for input condition was not significant, $F(1,15)<1$, nor did input condition interact with blocks, $F(5,75)<1$. The mean accuracies were 92% for the 2-input condition and 93% for the 6-input condition.

An analysis of the subjects' response times also failed to show differences between the 2- and 6-input conditions. The subjects responded faster, on average, to the 6-input gates (2.18 seconds) than to the 2-input gates (2.31 seconds), but this difference was not significant, $F(1,15)<1$. As one might expect, there was a significant speed-up over blocks, $F(5,75)=14.52$, $p<.001$; the means for the eight 50-trial blocks, beginning with block 1, were 2.77, 2.34, 2.19, 2.10, 2.02, and 1.90 seconds. The variables input condition and 50-trial block did not significantly interact, $F(5,75)=1.15$, $p>.34$.

In summary, the initial 216 trials of training brought the subjects to a high level of accuracy. The final test blocks showed that the subjects could maintain, and even improve, this accuracy when they were tested on the different gates at random. There was no indication that the 6-input gates were more difficult to learn than the 2-input gates.

Connectionist Learning Without Task Division

We also examined connectionist learning of the digital logic task using the backpropagation learning procedure. A software package developed by McClelland and Rumelhart (1988) was used to model the task. To find out how changing the number of inputs would affect learning, we modelled learning of 2-, 4- and 6-input gates.

The networks trained with backpropagation were feed-forward networks having either 6, 8 or 10 units in the input layer. Each network had 20 hidden units, and a single output unit. The input layer consisted of 3 units to encode gate type, 1 unit to encode negation, and 2, 4 or 6 units to encode the inputs (0's or 1's) to the gates. Figure 1 illustrates the network's configuration for learning the gates with 6 inputs. Different codes were used for the AND (100), OR (010), and XOR (001) gates, and the negation unit was set to 1 to represent the negated gates (NAND, NOR, and XNOR) and otherwise set to 0. The initial weights for the network were set to random values that varied uniformly between -0.5 and 0.5. The momentum parameter was set to 0.9. We tried a number of different learning rate parameters, and the simulations we report below used the parameters that yielded the fastest learning. These learning rate parameters were .1, .07, and .02 for the 2-, 4-, and 6-input networks respectively. The learning rates had to be reduced as the number of input units were increased to yield reasonably stable learning times.

Following the usual procedure for backpropagation, the networks were repeatedly presented with the complete set of patterns to be learned in cycles or "epochs." The networks were presented with patterns corresponding to all possible feature combinations for the gates and their inputs. Particular patterns in the 4- and 6-input simulations were repeatedly presented to the network within epochs to achieve the same proportion of 1 and 0 responses that subjects

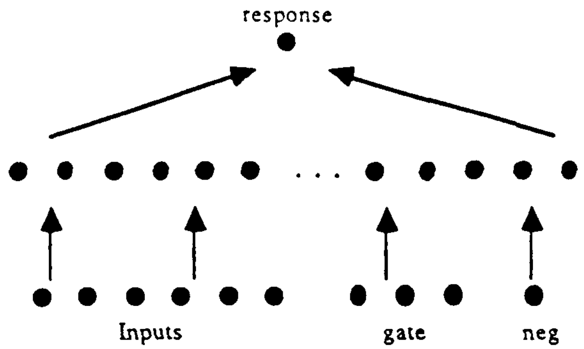

Figure 1. The configuration of the network that learned the 6-input gates without task division.

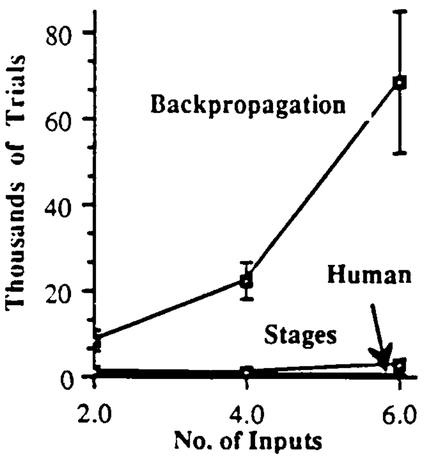

Figure 2. Trials to criterion for humans, backpropagation alone, and backpropagation with stages. The bars represent standard deviations.

had encountered in the experiment described above. The weights were adjusted after each pattern so that the network learned over epochs to respond to the patterns with the appropriate 0's and 1's.

Each network's accuracy was tested at 10-epoch intervals during learning by presenting the set of training patterns to the network while learning was turned off. A network's response was assumed to be a 1 if the activation of the output exceeded .5, and 0 if its activation was less than .5 (possible activation values varied between 0 and 1). Ten simulations were run for the different network configurations, each starting with different random weights.

Figure 2 shows the number of trials (number of epochs times number of patterns per epoch) needed for each network to learn to the criterion of 100% accuracy. This criterion was used because the network's behavior was deterministic; if the network was less than perfect it would always err on the same patterns. These systematic errors, which are uncharacteristic of our subjects who performed above 90% accuracy, were taken to mean that the network had not yet learned the task.

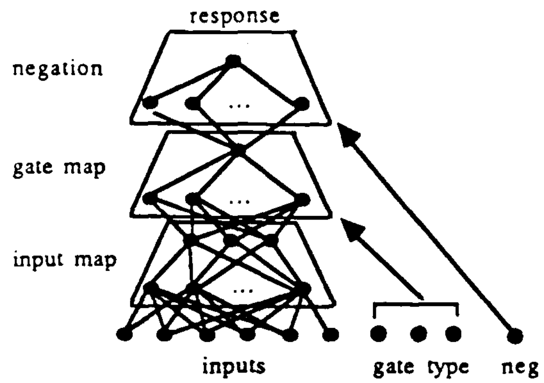

Figure 3. The configuration for the network that used task division to learn the 6-input gates.

As the complexity of the task increased, there was a substantial growth in the number of trials necessary to train the networks. Note that this growth contrasts dramatically with the lack of any complexity effect in the human data. This growth apparently resulted from the exponential increase in the number of patterns to be learned by the network; the number of patterns to be learned doubled with each additional input. There were 24, 96, and 384 patterns to be learned in the 2-, 4, and 6-input conditions respectively. Generalization of learning among the patterns was insufficient to hold down the learning time.

Connectionist Learning with Task Division

Human learning may scale well in our task because of the subjects' abilities to divide the task into component tasks. These component tasks can be separately focused on during both instructions and performance of the task. The subjects' prior knowledge allows them to be instructed on the rules that apply to the component task and would, even in the absence of explicit instructions, allow them to form hypotheses about which feature combinations might be important. Such task division and use of prior knowledge are, of course, standard features in many simulations of cognitive processes, e.g., Anderson's ACT* (1983). Furthermore, the notion of information processing stages has played a fundamental role in cognitive psychology. Much reseach has been designed to identify stages of processing and discover how they interact (e.g., Sternberg, 1969).

To examine how task division might speed up learning in our task, we used backpropagation to learn the individual component tasks in a modular network. Figure 3 illustrates how the units that coded the gate inputs, gate type, and negation were used as inputs to the modules. The figure also shows how the outputs from one module became the inputs to another module. The model had three modules, each containing a layer of input units, a layer of 10 hidden units, and a layer of output units. The first module (input map) was trained to recode 2, 4, or 6 inputs of 0's and 1's into codes representing either "all 0's", "all 1's", or "mixed." The second module (gate map) was trained to produce the correct responses (1 or 0) when given the recoded inputs and the codes for the gate types (AND, OR, XOR). The third module (negation) was trained to negate the output of the second module when negation was called for.

To assess total times for the model to learn the task, learning simulations were run for each module. Our results on learning times are based on 10 runs for each simulation. Each run was initialized to use a different set of random weights uniformly distributed between -.5 and +.5. For all modules, the momentum parameter was .9. The learning rate parameters for the input-map module were .5, .1, and .05 for the 2-, 4-, and 6-input conditions respectively. The

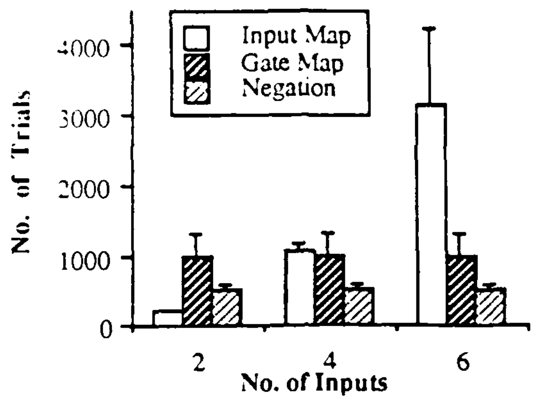

Figure 4. Trials to criterion as a function of subtasks and no. of inputs. The bars represent standard deviations.

learning rate parameters were .1 for the gate-map module and .5 for the negation module. These learning rate parameters were selected to enable rapid learning, but no major effort was taken to find the best parameters.

Figure 4 shows the mean number of trials needed to learn the component tasks for the different numbers of gate inputs. Figure 4 shows that recoding the input as 1's, 0's, and mixed requires substantially more trials as the number of inputs is increased. Assuming that learning can occur for all three modules during each trial, learning time would depend principally on the module that took the maximum number of trials to learn. This maximum value is plotted in Figure 2. It is clear from the figure that learning in this case scales considerably better than learning with backpropagation alone. It should be pointed out, however, that Figure 4 suggests that many more trials would be needed to learn gates with more than six inputs. If presented with more inputs, the subjects would probably adopt additional coding processes to cope with increasing complexity, as is thought to occur when subjects chunk visual stimuli into familiar configurations (Bartram, 1978).

Discussion

We have examined human and connectionist learning of a modestly complex problem. The human subjects learned the task very quickly, reaching 90% accuracy by the second block of distributed practice. There was no evidence of any problem of scaling in the human learning data, with both the 2- and 6- input conditions reaching an asymptote of 93% in 358 trials. Reaction times declined substantially over trials, with the 2- and 6- input functions showing equivalent learning rates. In an extended study of human learning of digital gates (Carlson et al. 1988a) subjects took about 500 trials per gate or 3000 total trials to bring their response times below .8 seconds. When responding in .8 seconds, subjects have apparently shifted to a strategy of direct associative retrieval of the output of each stage given its input (see Carlson et al., 1988). To acquire this skill of automatic retrieval in the digital-logic task, subjects require about 5 hours of practice distributed over several sessions.

In sharp contrast to human learning, connectionist learning without task decomposition required about 68,000 trials to learn the 6-input case. Assuming that humans take about 6 seconds per trial, about 110 hours would be needed to perform 68,000 trials. This is far more than the 5 hours humans actually required. Even of greater concern than this long learning time, is the poor scaling shown in learning. The network required about 6 times as many trials

to learn the 6-input as the 2-input case. The dramatic growth in the number of training trials suggests such a network could not learn an 8-input problem in the lifetime of a human.

Connectionist learning with task decomposition learned the 6-input case in about 3,200 trials and scaled fairly well, requiring 3 times as many trials as the 2-input case. The total number of trials compares reasonably well with the human performance, at least if we assume that the human connectionist processing is not well developed until humans can respond below 1 second. Connectionist learning with decomposition learned the 6-input case 21 times faster than without decomposition.

The above results suggest that combining rule-based and connectionist learning may provide the best of both types of computation. Initial rule-based learning (as in ACT* and SOAR) can search a problem space and decompose a task into subtasks in reasonable amounts of time. Processing in this rule-based mode is slow, serial, and effortful as is a human novice during the controlled-processing stage of skill acquisition (Shiffrin & Schneider 1977, Schneider & Detweiler 1987). Practice executing the rules allows connectionist learning to map the inputs to the outputs of each of the component tasks. The early rule-based processing decomposes a task so that smaller-scale tasks can be learned with connectionist procedures. This decomposition must identify the basic stages and the number of output states for each stage. Once tasks have been divided, connectionist learning need no longer perform gradient descent search in the power set of all possible connections, but rather has a more limited problem of mapping a small number of input states of each component task to a small number of output states for each component task. This use of task decomposition to make connectionist learning scale reasonably is an approach also advocated by Minsky (1988) to deal with the combinatoric explosion problem that occurs as task complexity increases.

Some readers might argue that our example provides an unfair test of connectionst learning and that our conclusions apply to only a limited set of tasks. We will briefly discuss four criticisms readers may have. *First, the problem chosen was a particularly difficult one for connectionist learning, since it included three levels of non-linearly separable problems (inputs, gates, negation).* We grant this, but it is a real task that humans have no difficulty performing if they are instructed. Learning combinatoric gates is still a toy problem and one that must be solved by any model of human learning. *Second, by instructing humans we gave away the answers.* We agree, but standard connectionist learning provides no mechanism for instruction. Since human learning can improve by many orders of magnitude with instruction, it is important to explore architectures that can benefit from instruction. *Third, different parameters or new learning algorithms may greatly speed learning in the present task, so that a connectionist procedure could learn the 6-input condition in a reasonable number of trials.* Perhaps, but the critical issue is whether new solutions will scale well. Task division and use of rules can always be used to reduce the scaling problem for any connectionist procedure, and it would be surprising if human learning would not make use of this property when learning new tasks. *Fourth, the present study shows that dividing tasks brings about faster learning, but there is no demonstration of how to implement the task decomposition in a parsimonious manner.* We are currently working on developing such an architecture.

We are developing a connectionist/control architecture (Schneider & Detweiler 1987, Schneider & Mumme 1988, Schneider & Oliver, 1988) that can implement rule-based learning and connectionist learning and that can benefit from instruction and task division. The architecture involves connectionist modules that transmit vector messages among modules. The control architecture uses an attentional gating mechanism that can modulate the transmission and reception of vectors among modules. Each module outputs information to the controller, indicating the degree of module activity and priority of its message. Controlled processing of the rules involves altering what messages are transmitted and compared in the network. For example, in digital-gate learning, the rule would be of the form "if all the input module vectors match the lexical vector module (which contains a 1); then transmit the "ALL1s" code to the

output of the input-coding module". Through changes in attentional gating, the network can be reconfigured to execute a process in as many stages as is required to perform the task. Intermediate states for each stage are represented not as specific units, but as random vectors.

Learning during the input-coding stage illustrates how rule-based and connectionist learning interact in the connectionist/control architecture. The instructions to the model indicate that the input code must be encoded in one of three critical states and all the inputs map to these critical states. The network generates three random-state vectors and associates those to their respective rules (e.g., All1s = A; ALL0s=B, MIXED=C). The random vectors are similar to the *gensym* operator in LISP programs. During practice, the rule-based performance correctly solves the problem by serially executing the rules. On each trial the input and output of each stage are correctly set via the rule-based processing (Schneider & Mumme, 1988). Connectionist learning alters the connection weights to directly map the input to the output without the use of the rule. As opposed to doing a gradient descent search through the connection space for all possible output codes, the network needs only to learn how to map the input states to the instructed output states.

As the connectionist/control architecture learns a task, processing shifts from sequential, rule-based to association-based processing. Each module associatively maps its input to the output and this process cascades over a number of stages. This connectionist processing has two important advantages over rule-based processing. First, it is faster, because information is retrieved associatively. Second, it is not as brittle as rule-based processing because the mutual constraint match property of connectionist mapping will map the input to its closest matching output. This may provide better generalization when the rule knowledge is ambiguous. The model follows the changes in human skilled performance as practice continues (Schneider & Detweiler 1987; Schneider & Mumme, 1988).

Summary

We have provided an illustration of the scaling problem exhibited by backpropagation when required to solve a modestly complex task. We have shown that humans, if they are given instruction on the digital-logic task, show no effect of scale when the number of inputs to be learned was increased. The humans learned the most complex task 220 times faster (in terms of trials) than the connectionist simulation. We also evaluated a model using a task decomposition exhibited by the human subjects. Connectionist learning of the decomposed tasks scaled reasonably in this model, learning 21 times faster than the model without task decomposition for the 6-input case. We speculated that hybrid architectures provide a superior processing environment than either purely rule-based or connectionst processing environments. The hybrid approach appears to scale well and to learn at rates comparable to humans.

Acknowledgments

We thank Mark Detweiler for his comments on a draft of this paper. This research was supported by the Army Research Institute, under contract No. MDA903-86-C-0149 and Personnel and Training Research Programs Psychological Sciences Division, Office of Naval Research under Contract Nos. N-0014-86-K-0107 and N-00014-86-K-0678.

References

Anderson, J. R. (1983). *The architecture of cognition*. Cambridge, MA: Harvard University Press.

Bartram, D. J. (1978). Post-iconic visual storage: Chunking in the reproduction of briefly displayed visual patterns. *Cognitive Psychology, 10*, 324-355.

Biederman, I., & Shiffrar, M. M. (1987). Sexing day-old chicks: A case study and expert systems analysis of a difficult perceptual-learning task. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 13*, 640-645.

Brooke, J. B., & Duncan, K. D. (1983). Effects of prolonged practice on performance in a fault-location task. *Ergonomics, 26*,379-393.

Carlson, R. A., Sullivan, M. A., & Schneider, W. (1988a). *Fluency of component judgments in causal network problems*. Unpublished manuscript.

Carlson, R. A., Sullivan, M. A., & Schneider, W. (1988b). *Practice and working memory effects in building procedural skill for causal judgments*. Manuscript submitted for publication.

Hinton, G. E. (1986). Learning distributed representations of concepts. In *Proceedings of the Eighth Annual Conference of the Cognitive Science Society* (pp. 1-12). Hillsdale, NJ: Erlbaum.

Laird, J. E., Rosenbloom, P. S., & Newell, A. (1986). Chunking in Soar: The anatomy of a general learning mechanism. *Machine Learning, 1*, 11-46.

McClelland, J. L., & Rumelhart, D. E. (1988). *Explorations in Parallel Distributed Processing: A handbook of models, programs, and exercises*. Cambridge, MA: MIT Press.

Medin, D. L., & Schaffer, M. M. (1978). Context theory of classification learning. *Psychological Review, 85*, 207-238.

Minsky, M., & Papert, S. (1988). *Perceptrons (Expanded Edition)*. Cambridge, MA: MIT Press.

Minsky, M. (1988). Preface: Connectionist models and their prospects. In D. Waltz and J. A. Feldman (Eds.), *Connectionist models and their implications: Readings from Cognitive Science* (pp. vii-xvi). Norwood, NJ: Ablex.

Rumelhart, D. E., Hinton, G. E., Williams, R. J. (1986). Learning internal representations by error propagation. In D. E. Rumelhart and J. L. McClelland (Eds.), *Parallel Distributed Processing* (pp. 318-364). Cambridge, MA: MIT Press.

Shiffrin, R. M., & Schneider, W. (1977). Controlled and automatic human information processing: II: Perceptual learning, automatic attending, and general theory. *Psychological Review, 84*, 127-190.

Schneider, W., & Detweiler, M. (1987). A connectionist/control architecture for working memory. In G. H. Bower (Ed.), *The Psychology of Learning and Motivation* (Vol 21, pp. 54-119). New York: Academic Press.

Schneider, W., & Mumme, D. (1986). *Attention, automaticity and the capturing of knowledge: A two-level architecture for cognition*. Unpublished manuscript.

Schneider, W., & Oliver, W. L. (1988). *An instructable connectionist/control architecture: Using rule-based instructions to accomplish connectionist learning in a human-time scale*. Manuscript in preparation.

Sternberg, S. (1969). The discovery of processing stages: Extensions of Donder's method. *Acta Psychologica, 30*, 276-315.

Volper, D. J., & Hampson, S. E. (1986). Connectionist models of boolean category representation. *Biological Cybernetics*, 54, 393-406.